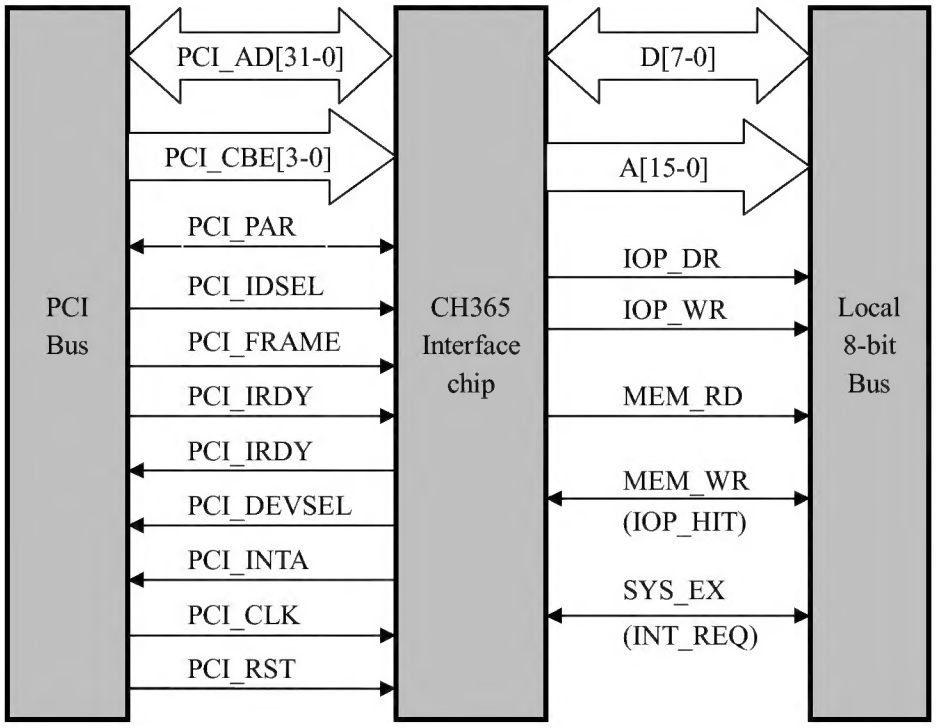
PCI Bus Interface Chip CH365

Datasheet
Version: 1D
http://wch.cn

1. Introduction

CH365 is a general interface chip based on PCI bus, supports I/O port mapping, expansion ROM and interrupt. CH365 converts the 32-bit high-speed PCI bus into an easy-to-use 8-bit active parallel interface which is similar to ISA bus, for making low-cost computer board card based on PCI bus, and upgrading the original board card based on ISA bus onto PCI bus. Compared with other mainstream buses, PCI bus has higher speed, better real-time and controllability, so CH365 is suitable for high-speed real-time I/O control card, communication interface card, data acquisition card, electron disk and expansion ROM etc.

The figure below shows its general application block diagram.

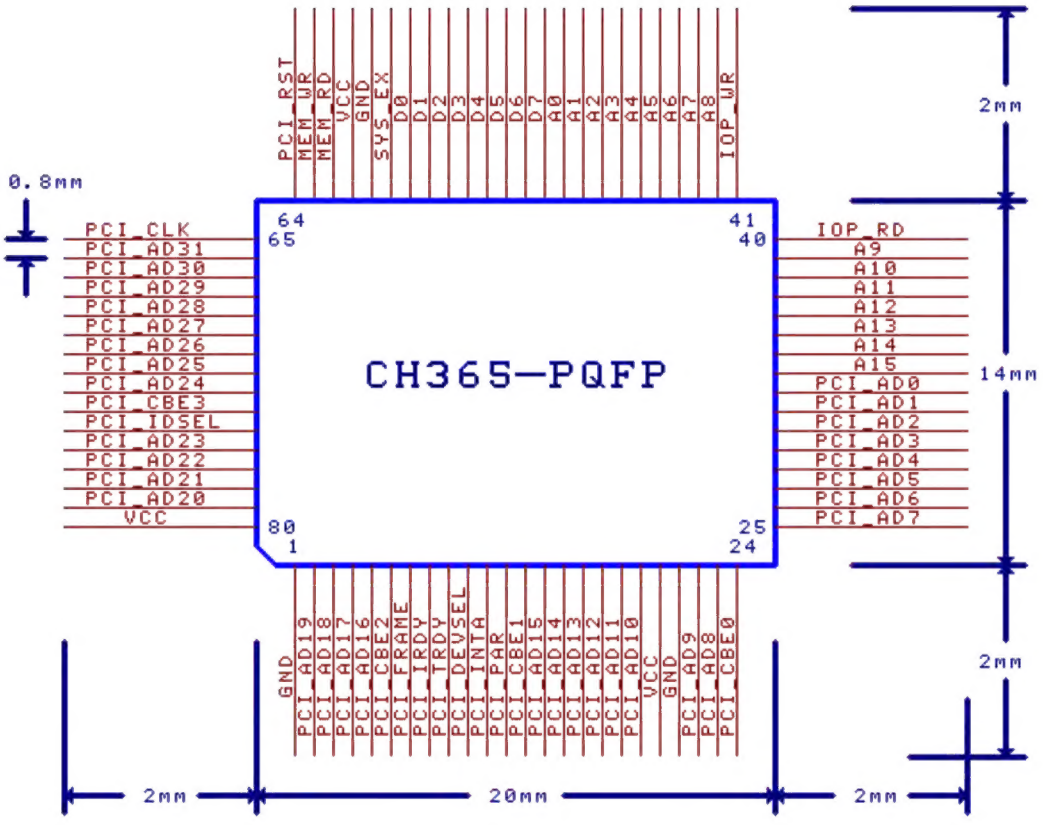


2. Features

- Realizes the slave device interface based on 32-bit PCI bus.
- Converts into active parallel interface: 8-bit data, 16-bit address, I/O read and write, memory read and write.
- Can set device identification (Vendor ID, Device ID, Class Code, etc.) of PCI board.
- Supports write and read I/O port or memory in byte, character and double characters as unit.
- Test speed of no burst access can reach 7MB/s, and Read and write pulse width is selectable from 30nS to 240nS.
- Distributes I/O base address automatically, and supports I/O port up to 240 bytes.
- Supports local hardware fixed address function, freely chooses address, realizes I/O port at the appointed address.
- Directly upgrades the I/O board card of ISA onto PCI bus without modifying the interrelated software of original ISA card.
- Supports directly mapping 32KB memory SRAM or expansion ROM (Boot ROM).

- Supports 64KB or 128KB memory or expansion ROM without adding the external components.
- Supports the expansion ROM without hard disk booting and Flash-memory updates on line.
- Provides the sub-program library BRM of expansion ROM application which is used to display user interface and process data under BIOS.
- Support local interrupt request of low-level active and interrupt sharing.
- Provides two-wire serial host interface, which can connects with the similar component of 24C0X EEPROM.
- Built-in 4uS to 1mS hardware timing unit, used as a delay reference during software running.
- PQFP-80 package with the pitch of big pin (0.8mm) that is compatible with CH361P pins which can directly replace.
- Drive supports Windows 98/ME/2000/XP/7/8/10 and provides application layer API through DLL.
- NO driver required for CH365 itself work and upgrading the ISA card.
- Multiple patented technologies, easy-to-use and low-cost.

3. Package



Package	Width Of Plastic	Pitch Of Pin		Instruction Of Package	Ordering Information
PQFP-80	20mm x 14mm	0.8mm	31.5mil	Small outline PQFP-80 pin patch	CH365P

4. Pin Out

4.1. Power Line

Pin No.	Pin Name	Type	Pin Description
20,61,80	VCC	Power	Power supply voltage

1,21,60	GND	Power	Ground

4.2. PCI Bus Signal Line

Pin No.	Pin Name	Type	Pin Description
64	PCI_RST	Input	System reset signal lines, active low
65	PCI_CLK	Input	System clock signal lines, active rise edge
2~5 14~19 22, 23 25~32 66~73 76~79	PCI_AD31~PCI_AD0	Tri-status output and input	Address and bi-directional data multiplexed signal lines
6,13 24,74	PCI_CBE3~PCI_CBE0	Input	Bus command and byte enable signal lines
12	PCI_PAR	Tri-status and bi-direction	Odd-even Parity signal line
75	PCI_IDSEL	Input	Initializing device select signal line, active high
7	PCI_FRAME	Input	Frame cycle start signal line, active low
8	PCI_IRDY	Input	Originating device ready signal line, active low
9	PCI_TRDY	Tri-status output	Target device ready signal line, active low
10	PCI_DEVSEL	Tri-status output	Target device select signal line, active low
11	PCI_INTA	Tri-status output	INTA interrupt request signal line, active low

4.3. Local Signal Line

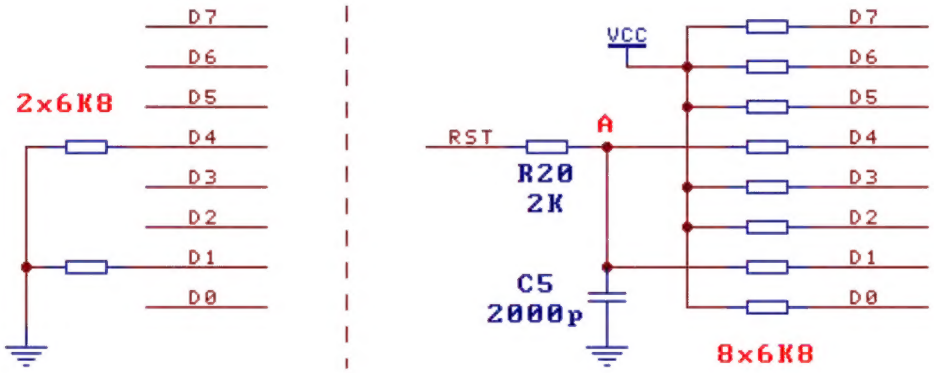
Pin No.	Pin Name	Type	Pin Description
51~58	D7~D0	Tri-status output and input	8-bit bi-directional data signal line, built-in pull-up resistor, D7 is the signal line SDA of 2-wire serial interface simultaneously
33~39 42~50	A15~A0	Output	16-bit address signal line, A15~A10 can control output independently, A15 can set as the signal line SCL of 2-wire serial interface
40	IOP_RD	Output	Read strobe/ enable of I/O port, active low
41	IOP_WR	Output	Write strobe/ enable of I/O port, active low
62	MEM_RD	Output	Read strobe/ enable of memory or expansion ROM, active low
63 Multiplex	MEM_WR	Output	Write strobe/ enable of memory or expansion ROM, active low
	IOP_HIT	Input	Local hardware fixed address request, active low, built-in pull-up resistor
59 Multiplex	SYS_EX	Output	Signal line of independent control output, can set as the signal line SCL of 2-wire serial interface
	INT_REQ	Input	Local interrupt request input, active low, built-in pull-up resistor

5. Operation Mode Setting

For providing more usable functions without adding pins, CH365 can multiplex some pins through the "Operation Mode Setting" for function selection. The detailed method as follow: making the Local 8-bit data lines D7~D0 are the required high or low level by pull-up or pull-down mode. After CH365 is reset, set the operation mode and parameters according to the default status of these signal lines. When these signal lines are driven as an 8-bit data bus, because the driving current is not less than 1mA, the pull-up or pull-down

won't affect the driving of the data bus. In addition, CH365 sets the operation mode and parameters within once time in the 1us after being reset. If the drive ability of peripheral equipment is low or it is the OC collector driver, the pull-down is realized in the short time after reset, and shield pull-down or transformed into pull-up in other time.

The following figures are the two circuits for the operation mode setting. CH365 has connected a 40kΩ weak pull-up resistors to the 8-bit data lines D7~D0 inside the chip. The data line is low level when it is connected with pull-down resistor outside the chip. Otherwise, it is high. The circuit on the left in the figure is suitable for general conditions. D4 and D1 are pulled down to low level by the 6.8kΩ resistor, so the value of data lines D7~D0 is 11101101(binary), because the resistance value of the pull-up or pull-down resistor is high, CH365 and peripheral equipments will not affect the driving of the data bus. When the drive ability of peripheral equipments is lower than 1mA or it is the collector driver, you can consult the circuit on the right in the figure. In the figure, RST is the PCI reset signal. Because of the capacitor C5, the point A will continue to maintain a low level for several ms after RST reset is completed. CH365 completes the operation mode setting during this period. The value of data lines D7~D0 is 11101101 on the right circuit in the figure.



The following table shows the value of the operation modes and parameters (1=high, 0=low). For example, D1=0, define the PCI card ID; D3=0, active interrupt; D4=0, active local hardware fixed address; D3 and D4 can't be 0 at the same time.

Data Bus	Instruction Of Data Bus	Value =0	Value =1
D0	Set default value of A15 after system reset	A15 =0	A15 =1
D1	Select PCI Vendor ID and Device ID	External ID	Default ID of CH365
D3	Select function of multiplexed pin 59	INT_REQ	SYS_EX
D4	Select function of multiplexed pin 63	IOP_HIT	MEM_WR
other	Provided to product manufacture, the status of data line can be read from PCI configuration space 41H. For example, optionally connect pull-down resistors on D2, D5, D6 and D7 which can identify the board function for application.		

6. Registers

6.1. Basic Specification

6.1.1. Abbreviation of Attribute: R=read-only, W=read and write, S=read-only but can be set in advance,….= ellipsis.

6.1.2. Numerical system of data: If it ends with H which is a hexadecimal number. Otherwise, it is a binary number.

6.1.3. Wildcard character and attribute of numeric value: r=reserved (disabled), x=any value, ...=

ellipsis.

6.2. Configuration Space 0FFH-00H

Class	Address	Register Name	Register Attribute	Default Value After System Reset
Standard PCI configuration space	01H-00H	Vendor ID	SSSS	4348H
	03H-02H	Device ID	SSSS	5049H
	05H-04H	Command	RRRRRRRRRRRRRRWW	0000000000000000
	07H-06H	Status	RRRRRRRRRRRRRRRR	0000010000000000
	08H	Revision ID	SS	10H
	0BH-09H	Class Code	SSSSSS	100000H
	0FH-0CH		SSSSSSSS	00000000H
	13H-10H	I/O Base Address	RRRRRRRRRRRRRRRR WWWWWWWWRRRRRRRR	0000000000000000 0000000000000001
	17H-14H	Memory Base Address	WWWWWWWWWWWWWWWW WRRRRRRRRRRRRRR	0000000000000000 000000000000x000
	2FH-18H		SSSS….SSSS	0000….0000H
	33H-30H	ROM Base Address	WWWWWWWWWWWWWWWW WRRRRRRRRRRRRRRW	0000000000000000 0000000000000000
	3BH-34H		SSSSSSSSSSSSSSSS	0000000000000000H
	3FH-3CH	Interrupt Line & Pin	RRRRRRRRRRRRRRRR RRRRRRRRWWWWWWWW	0000000000000000 0000000x00000000
Control register and miscellaneous	40H	Chip control register	RRRRRWWW	rrrrr00x
	41H	8-bit bus D7-D0 Output ports	RR	xxH
	42H	Chip status register	RRRRRRRR	xxrxrxrx
	43H	Reserved	(Disabled)	(Disabled)
	Other	4FH-44H alias 43H-40H, reserved 0FFH-50H, only-read 00H		

6.3. I/O Space 0FFH-00H

(The actual address of the register is the I/O base address plus the offset address in the table)

Class	Offset Address	Register Name and Instruction	Register Attribute	Default Value After Reset
Port	0EFH-00H	Standard local I/O port	WW	Connect to I/O device
Memory port	0F0H	A7-A0 address set register	WWWWWWWW	xxxxxxxx
	0F1H	A15-A8 address set register	WWWWWWWW	000000xx
	0F2H	Reserved	(Disabled)	(Disabled)
	0F3H	Memory data access register	WW	Connect to memory
2-wire serial interface	0F4H	Data access register	WW	xxH
	0F5H	Control and status register	WRRRRRRW	00000000
	0F6H	Address set register	WW	00H

	0F7H	Device address and command register	WW	00H
Control registers and miscellaneous	0F8H	Chip control register	RRRRRWWW	rrrrr00x
	0FAH	Read and write speed control register	WRRWRWWW	0rr0r111
	0FCH	Hardware cycle count register	RR	xxH
	other	Reserved	(Disabled)	(Disabled)

6.4. Register Bit Instruction

Register Name	Address	Attribute	Instruction For Use Of Bit (Default Value)	Value=0	Value=1
Chip control register (configuration space 40H) (I/O space offset 0F8H)	Bit 0	W	Set input value of A15	Low	High
	Bit 1	W	Set input value of SYS_EX	Low	High
	Bit 2	W	Set active status of INTA interrupt	No interrupt	interrupt
Chip control register (configuration space 42H)	Bit 0	S	Current selection of PCI device	External ID	Default ID
	Bit 2	S	Local hardware fixed address function	Disabled	Enabled
	Bit 4	S	Internal Boot-ROM function	Disabled	Enabled
	Bit 6	S	SYS_EX output signal	Disabled	Enabled
	Bit 7	S	Enable status of interrupt function	Disabled	Enabled
2-wire serial interface control and status register （I/O space offset 0F5H）	Bit 0	W	Operation status of 2-wire serial interface	Completed	Operated
	Bit 7	W	Select SCL output signal	A15	SYS_EX
Read and write speed control register （I/O space offset 0FAH）	Bit 2 Bit 1 Bit 0	W W W	Bit2~0 set read and write pulse width，interval is 30nS; Bit4 is 0, 000~111 correspond to 30nS~240nS; Bit4 is 1, 000~111 correspond to 0nS~210nS		
	Bit 4	W	Data and address output building time	15nS	45nS
	Bit 7	W	Memory space pre-stored control bit	Disabled	Enabled

7. Function Descriptions

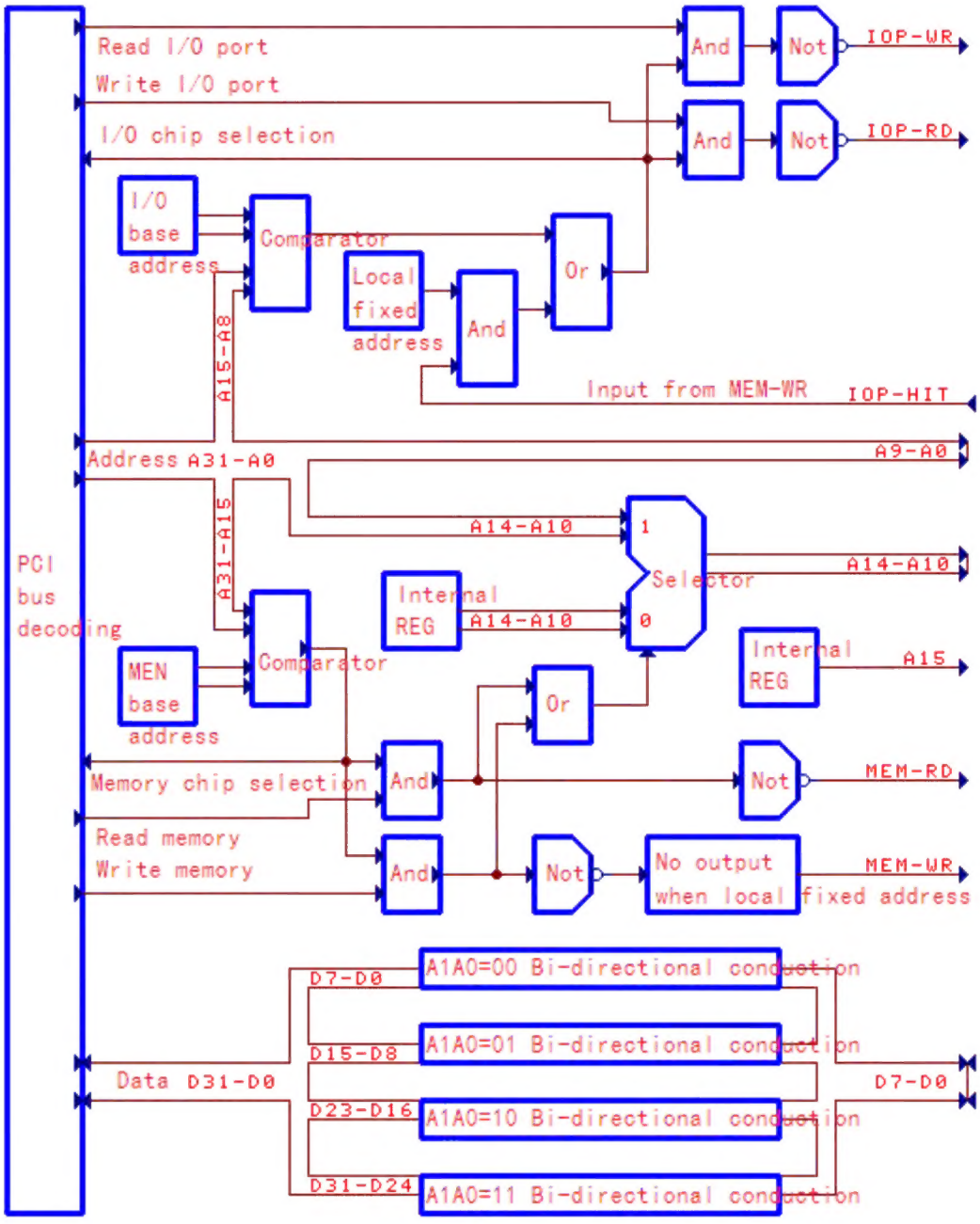
7.1. Space Mapping

There are three types of spaces in the PC: memory space, I/O space, and configuration space. The memory space, mainly including memory, video memory, expansion ROM and device buffer, etc., which is generally used to store large amounts of data and exchange data blocks. The I/O space, mainly including the control register and status register of the device, which is generally used to control and query the working status of the device and exchange a small amount of data. The configuration space is mainly used to provide basic information of the device to the system, and to accept the system's control and query for the device's global status.

To avoid address conflicts, PCI bus requires that the occupied address of each device can be relocated. Relocation is realized by the base address register of the configuration space of the device. Normally, the base address register of each device is always assigned to a different base address by the BIOS or OS, so that each device is separately mapped to a different address range. When needed, the application can also modify the base address by itself.

CH365 storage space occupies 32K bytes, the offset address is 0000H~7FFFH, all can be provided to extern al devices, the actual address is the memory base address plus the offset address. The I/O space of CH365 oc cupies 256 bytes, except for the self-use register of CH365, it also can provide 240 bytes for external device. The offset address is 00H~EFH, the actual address is the I/O base address plus the offset address.

7.2. Internal Structure and Signal Line



The figure above is the main structure of CH365. After CH365 decodes various signals of PCI bus, it generates internal data buses D31~D0, internal address buses A31~A0, read I/O port signal, write I/O port signal, read memory signal and write memory signal, etc. The transmission direction of each signal has been marked in the figure.

The signal on the right side of the structure diagram refers to the external pin provided by CH365 to the local terminal. The address lines A15~ A0 are used to provide offset addresses relative to the base address, and the data buses D7~D0 are used to input data during a reading operation, and are used to output data during a writing operation. IOP_RD is used to provide the I/O read strobe pulse signal, IOP_WR is used to provide

the I/O write strobe pulse signal, MEM_RD is used to provide the memory read strobe pulse signal, MEN_WR is used to provide the memory write strobe pulse signal, and the read and write strobe pulse signals of the above pins are all active at low level. The address line, data bus and the read and write strobe signal line provided by CH365 which are similar to the signal line of ISA bus, so it is very suitable for upgrading ISA board onto PCI bus. The figure indicates that the read and write strobe signals provided by CH365 have been controlled by chip selection inside the chip, and the read and write strobe signals outputted by CH365 are only valid within its base address mapping range, so the chip selection decoding is not required for the external device.

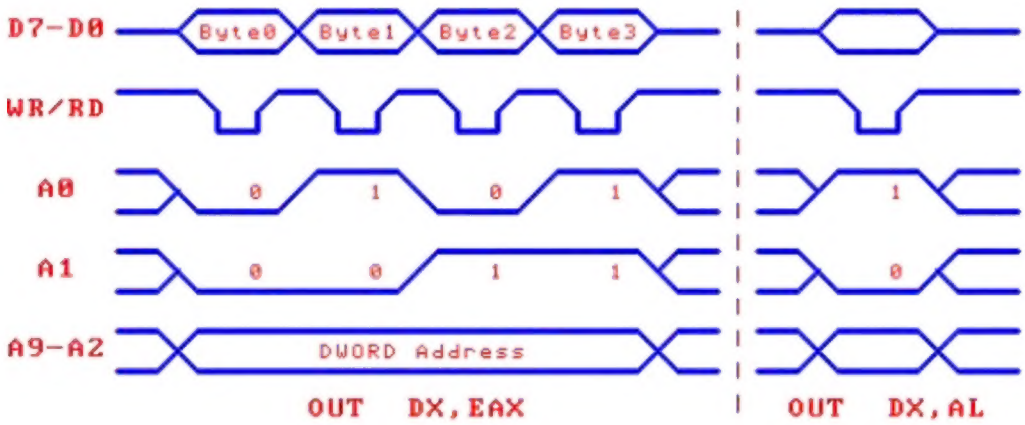
During the I/O read and write operation, A7~A0 of CH365 output the offset address of I/O port. The effective offset address range provided to the external device is 00H~EFH, and the external device can further decode A7~A0 to generate a secondary chip selection signal. During the I/O read and write operation, A15~A0 of CH365 keep unchanged, but it can be set to low or high in advance by the internal registers, and A9~A8 output address of PCI bus. If using local hardware fixed address function, which should decode the A9~A0 of CH365. CH365 requests local fixed address through IOP_HIT, it realizes the I/O port fixed address ranged from 000H to 3FFH, which is compatible with ISA bus.

During the memory read and write operation, A14~A0 of CH365 output the offset address of memory which provide the valid offset address ranged from 0000h to 7FFFH to peripheral devices. During the memory read and write operation, A15 of CH365 keep unchanged, but it can be set to low or high in advance by the internal registers which is used to expand address line or select page for memory. Because expansion ROM is one kind of memory, the operation mode and timing are same to memory.

7.3. Data Width

CH365 supports PC program to read and write I/O port or memory in single byte, double byte (character) and four bytes (double characters) as unit. During the continuous multi-byte read and write operation, After CH365 reads and writes a byte of data every time, it will automatically add 1 to the offset address to point to the offset address of the next byte. PC only need to execute a read and write four-byte command, CH365 will automatically decompose it into four continuous read and write operations of 8-bit data. Therefore, In the PC view, CH365 provides the 8-bit, 16-bit and 32-bit data width. In fact, the operation efficiency is higher and the whole data exchange speed is faster when it is 32-bit data width.

The left figure is the wave of assemble command “OUT DX EAX”. DX is the arbitrary double characters boundary address in the I/O base address range of CH365. For example, DX=IoBaseAddr+4; the right is the wave of assemble command “OUT DX AL”, DX is the arbitrary address in the I/O base address range of CH365. For example, DX=IoBaseAddress+1, which is correspond to the C-language command “outputportb (IoBaseAddr+1,value)”. I/O read operation and memory read and write operation are similar with I/O write operation. If PC executes a command of 32-bit data width, CH365 will automatically generate four continuous operations of 8-bit data, so the data transmission efficiency is higher than 8-bit data width.



7.4. Example Description

Designing a PCI board card similar to the printer port based on CH365. The design stipulates that the I/O offset address 00H of the board is the data port, the offset address 01H is the status port, and the offset address 02H is the control port. When it is inserted into a PC, the board card may be assigned with an I/O base address 9500H, then the actual I/O address of the data port is 9500H, the I/O address of the status port is 9501H, and the I/O address of the control port is 9502H. Distinction of each port is achieved by address decoding of A7~A0 of CH365. If other ports are not required, only A1~A0 can be simplified decoding.

If two identical boards are inserted into the PC, the second board card will also be automatically assigned with an I/O base address by the system, but it will not be the same as the I/O base address of the first board card. If the I/O base address of the second board card is C700H, the actual I/O address of the control port of the second board card is C702H, so that two identical PCI board cards have different I/O port addresses, avoiding I/O address conflicts.

The board card designer and related application know the offset address of each port in advance, but they cannot know the I/O base address in advance, so the application needs to know the I/O base address of current board card through the I/O base address register of configuration space of the board card before the application performs I/O operation on the PCI board card. Then, calculating the actual I/O address of each port is the I/O base address plus the offset address of each port. Finally, according to the actual I/O address, performing I/O operation on each port.

Memory is similar with I/O. There is a high-speed data exchange example of CH365 connected with 32KB double-port SRAM. If the memory base address of CH365 is distributed to E3050000H, PC program reads and writes the physical address range from E3050000H to E3057FFFFH which is reading and writing the double-port SRAM. Note, the actual PC program reads or writes dummy address not the physical address. In addition, if reading and writing the memory under DOS, the memory base address must be set below 1MB, such as 000D0000H or 000D8000H.

The following is an example of corresponding reading and writing process.

① Writing data 5AH to the control port, which is correspond to the C language program "outportb(0x9502,0x5A)". After execution, the address lines A7~A0 of CH365 output the offset address 02H of the control port (address 9502 is decomposed into base address 9500H and offset address 02H, and CH365 only outputs the offset address, not base address). CH365 data lines D7 ~D0 output 5AH, meanwhile, IOP_WR outputs a low-level pulse. The pulse width is pre-set by the read and write speed control register of CH365, and the default is 240nS.

② Reading data from the data port and status port, which is correspond to the C language program "inport(0x9500)". The low byte of the returned result which is the data read from the data port, and the high byte which is the data read from the status port. After execution, the address lines A7~A0 of CH365 firstly output the offset address 00H of the data port. Meanwhile, IOP_RD outputs the first low-level pulse. The external device outputs data to the data buses D7~D0; then, the address lines A7~A0 of CH365 output the offset address 01H of status port. Meanwhile, IOP_RD outputs the second low-level pulse. The external device outputs the data onto the date buses D7~D0.

③ Memory is similar with I/O, but there are two differences: Firstly, the address lines A14~A0 of CH365 output 15-bit offset address, and only A7~A0 of I/O output 8-bit offset address. Secondly, using the MEM_RD pin to output the read control signal instead of IOP_RD pin, and using MEM_WR pin to output write control signal instead of IOP_WR pin. This can enable external device to distinguish memory read and write operations rather than I/O port.

8. Parameters

8.1 Absolute Maximum Ratings

(Critical state or exceeding maximum can cause chip to not work or even be damaged)

Name	Parameter Description	Min.	Max.	Unit
TA	Operating ambient temperature	-20	70	℃
TS	Storage ambient temperature	-55	125	℃
VCC	Supply voltage (VCC connects to power, GND to ground)	-0.5	6.5	V
VIO	Voltage of input or output pins	-0.5	VCC+0.5	V

8.2 Electrical Parameters

(Test Conditions: TA=25℃, VCC33=5V, exclude pins connected to PCI bus)

Name	Parameter Description	Min	Typ.	Max	Unit
VCC	Supply voltage (refer to the below note)	3.3	5	5.5	V
ICC	Operating supply current	10	20	50	mA
VIL	Input low voltage	-0.5		0.8	V
VIH	Input high voltage	2.0		VCC+0.5	V
VOL	Output low voltage (4mA draw current)			0.5	V
VOH	Input high voltage (4mA output current)	4.5			V
IIN	Input current of the input without pull-up			10	uA
IUP	Input current of the input with pull-up		50		uA
RUP	Resistance value of pull-up resistor (not linear equivalent value)	40	60	100	KΩ

Note: the actual sustainable input voltage of CH365 is the supply voltage plus 0.5V. For example, when CH365 works at 3.3V, input voltage provided by peripheral equipments must not exceed 3.8V. When power voltage of CH365 is lower than 4V, the basic frequency of PCI bus must not higher than 33MHz, in other word, CH365 can't exceed the basic frequency.

8.3. Timing Parameters

(Test Conditions: TA=25℃, VCC33=5V, FCLK=33.3MHz, refer to the figure)

Name	Parameter Description	Min	Typ.	Max	Unit
FCLK	CLK input frequency (basic frequency of PCI bus)	0	33.3	40	MHz
TEN	Low-level pulse width of IOP_RD、IOP_WR、MEM_RD、MEM_WR read or write strobe	30	Optional 30~240	Default value 240	nS
TENS	High-level interval width of IOP_RD、IOP_WR、MEM_RD、MEM_WR continuous multi-byte strobes	Default value 30	Optional 30 or 60	60	nS
TAS	Address A15~A0 output building time	12	Optional 15 or 45		nS
TAH	Address A15 -A0 output keeping time	12	15		nS
TDS	Data D7-D0 output building time	12	Optional 15 or 45		nS
TDH	Data D7-D0 output keeping time	12	15		nS
TIS	Data D7-D0 input building time	15			nS
TIH	Data D7-D0 input keeping time	0			nS

TINT	Pulse width of INT_REQ interrupt request	80	100		nS
TSCL	SCL input frequency (basic frequency of 2-wire interface)	FCLK / 128 = 260 KHz			KHz
T12C0	SDA low time when start and end operation		3.84		nS
T12C1	SDA high time when start and end operation		3.84		nS
T12CS	SDA data output building time		1.92		nS
T12CH	SDA data output keeping time		1.92		nS

① Read and write strobe pulse width TEN is set by the bit 2~0 of read and write speed control register. Taking the 30ns of PCI bus clock cycle as the interval, it can be selected between 30ns-240ns and the error is 10%.

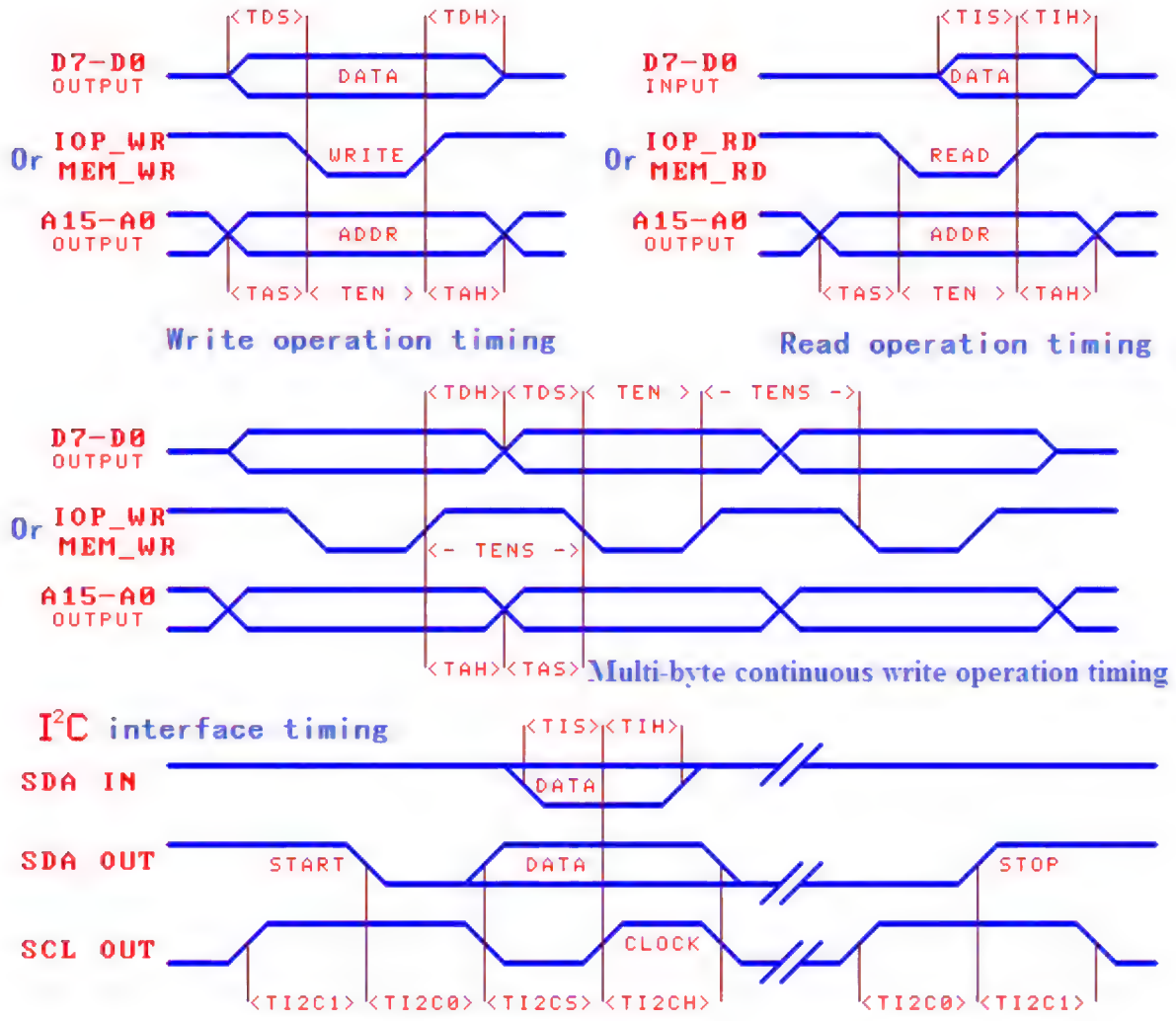
② In the multi-byte continuous read and write operation, the read and write strobe pulse interval is set by bit 4 of read and write speed control register. The interval is 30ns when the bit is 0, it is 60ns when the bit is 1, and the error is 10%. In the single byte operation, the minimum is 150ns, but it usually is set above 400ns. If CH365 is connected with double-port SRAM or peripheral circuits whose read and write speed is slower than 25ns, the interval is advised as 60ns.

③ Address output building time TAS and data output building time TDS are the advance time that the address and data output valid correspond to read and write strobe pulse falling-edge. TAS and TDS are set by bit 4 of read and write speed control register. The interval is 15ns when the bit is 0, the interval is 45ns when the bit is 1, and the error is 10%.

④ For read operation, CH365 sampled data from 8-bit data bus D7~D0 at the read strobe pulse rising-edge, so the external device should send the valid data to data bus before the read strobe pulse rising-edge. Data building time TIS is the advance time that data valid sent from external device to the data bus D7~D0 correspond to the read strobe pulse rising-edge.

⑤ Interrupt request pulse width TINT is the width that ensure CH365 receives the interrupt request low level. If the time of INT_REQ at low is too short, CH365 may not receive the interrupt.

⑥ 2-wire serial interface timing: SDA is input and output bi-directional port, CH365 outputs data to SDA at CLK failing-edge, and sampled input data of SDA at SCL rising-edge. If the 2-wire serial interface is idle, SCL keeps low, SDA tri-status output is disabled, but keep high because of internal pull-up resistor.

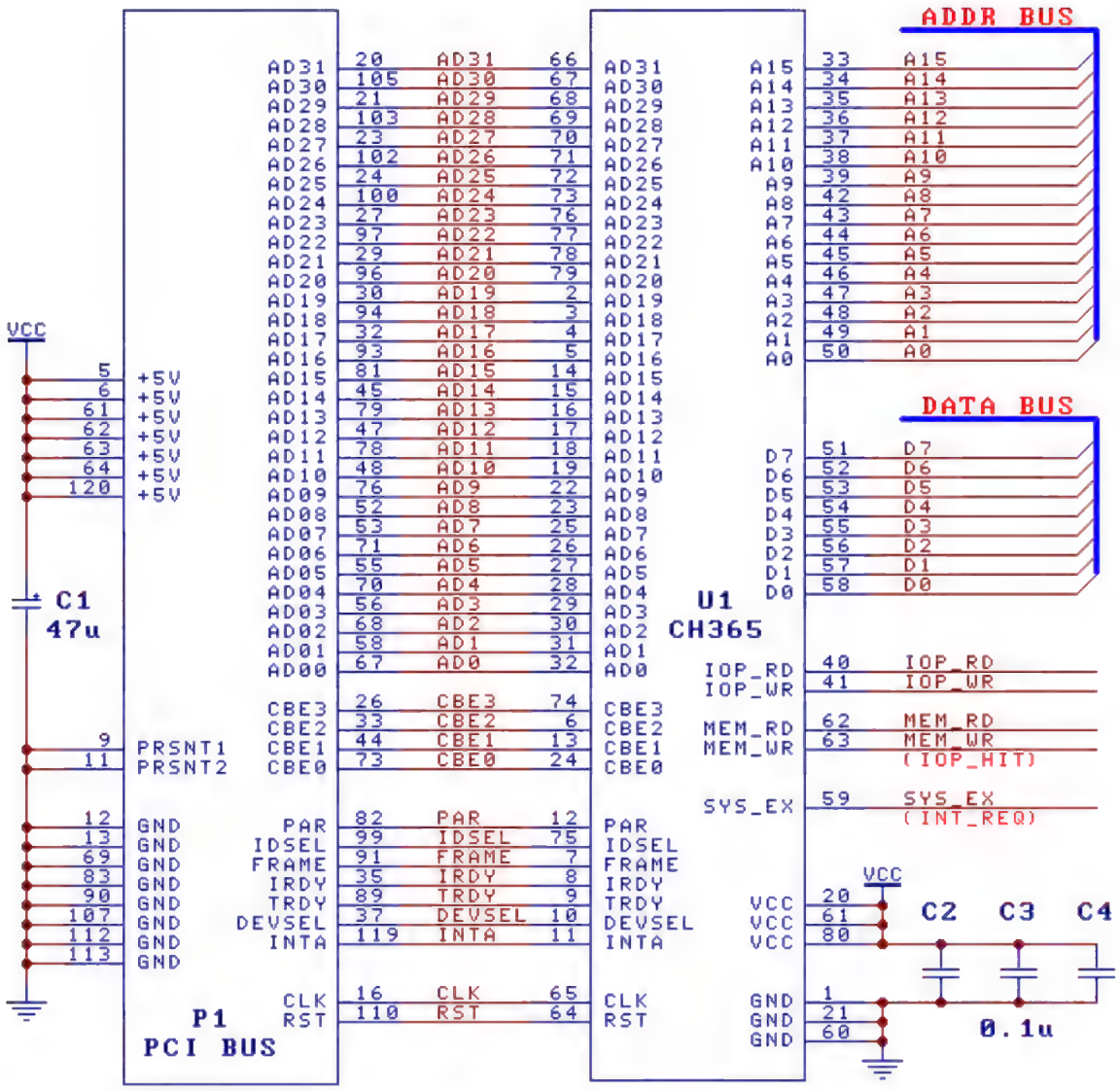


9. Applications

9.1. Connect to PCI Bus

This is the basic circuit that CH365 connects to PCI bus, capacitor C1~C4 are used for power decoupling, capacitor C2~C4 with a capacity of 0.1uF is a monolithic or high-frequency ceramic capacitor, and the number is not less than 3, which are connected in parallel nearby to the three pairs of power pins of CH365 respectively. The power lines can be selected freely which connected with PCI bus and the number is not less than 4.

CH365 is the high frequency digital circuit, the signal impedance matching should be considered. Please refer to PCI bus specification when designing the PCB board. It is recommended that the length of PCI signal line of CH365 is less than 35mm, line is arc or 45 degree rather than right-angle or sharp-angle as much as possible, signal line should be distributed on the component surface and leave a larger area connected with ground or covered with copper on the back of PCB. Keeping the length of PCI clock line CLK between 50mm and 65mm as much as possible, and it is not close with another signal line. It is suggestion that the both sides of CLK and the back of PCB should be grounded or covered with copper to avoid interference from surrounding signal lines.



9.2. Connect to Memory (left figure below)

CH365 connects with memory U2 (SRAM2256) through MEM_RD and MEM_WR. Before reading and writing the memory U2, the computer ought to read the memory base address register of CH365 configuration space. Then the base address added offset address to get actual physical address of U2 in the PC memory. Finally read and write the content of U2 through the actual physical address or the dummy address converted from physical address.

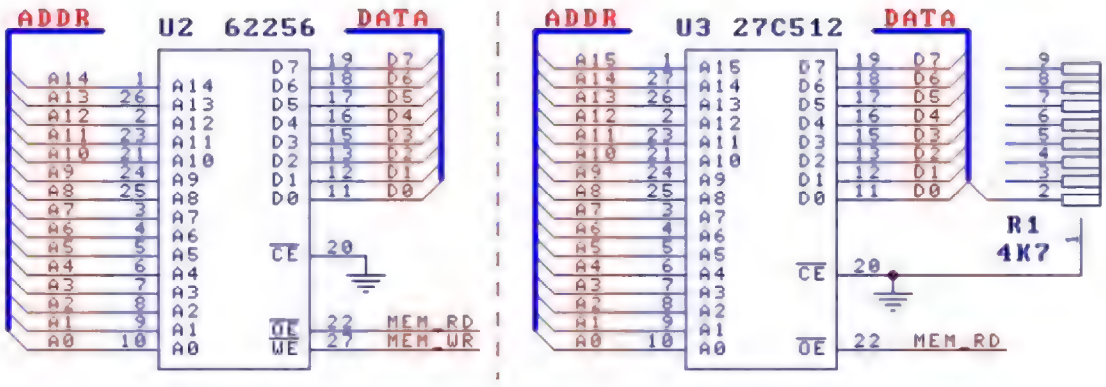
The C-language WDM and DLL program based on CH365 as follow:

```
UCHAR    mByte;
// data unit，used to save data read from the memory or to be written to memory
mPCH365_MEM_REG    mMemBase;
// memory base address，the actual data unit address is equal to the base address plus the offset address
CH365OpenDevice ( TRUE, FALSE );
// open CH365 device，similar with the file operation that opening it before using
CH365GetMemBaseAddr ( &mMemBase );
// get memory base address of CH365
CH365ReadMemByte ( & mMemBase -> mCh365MemPort[0x1234], &mByte );
// the above operation, get a byte data from the memory address 1234H
CH365WriteMemByte( & mMemBase -> mCh365MemPort[0x2E0C], mByte + 0x76 );
```

```
//the above operation, add the 76H to the previously read data, then write it to the memory address
  2E0CH

CH365CloseDevice( );
// all the operations are over, close Ch365 device before exit the application
  In DOS or PC without OS, the assembler program as following:
MOV   AX, 0B10DH          ; write the PCI configuration space in the unit of double-character
MOV   BX, CH365_PCI_BUS_DEV_ADDR   ;PCI address of CH365 board,
                                    namely, bus/device/function number
MOV   ECX, 000D0001H      ; set memory at D000H which is idle in the PC
MOV   DI, 0014H           ; offset address PC_BASE_ADDR1 of memory base address register
INT   1AH                 ; set memory base address to active memory space;
                          ; automatically attributed memory base address is above 1MB, because
                            DOS can't locate in it, it need to modify the base address
MOV   AX, 0D000H
MOV   ES, AX              ; set address at D000H
MOV   AL, ES:[1234H]      ; get a byte data from the memory address 1234H
ADD   AL, 76H
MOV   ES:[2E0CH], AL      ; add 76H to the read data, then write it to the memory address 2E0CH
```

If you replace ordinary SRAM with dual-port SRAM, CH365 can exchange data with external MCU or DSP through the dual-port memory. If the width of the read and write strobe pulse of CH365 is set to 30ns and exchange data in unit of double characters, the test speed of data transmission is 7MB/s.



9.3. Connect to Expansion ROM (right figure above)

CH365 connects with ROM U3 (27C512) via MEM_RD. CH365 supports the 32KB or 64KB EPROM and FLASH-Memory. IF SYS_EX is used for A16 address bus, the maximum capacity can reach 128 KB. In general, CH365 directly supports 32KB expansion ROM (the capacity of 27C256). CH365 supports the 64KB ROM by controlling the A15 address line in expansion ROM program. The selectable pull-down resistor R1 is used for setting operation mode of CH365. In the figure D0 is connected with pull-down resister, so A15 is low after system reset and select the low 32KB of U3 (offset address is 0000H ~7FFFH). Resetting the A15 by writing bit 0 of the chip control register when you need to read the high 32KB of U3 (offset address is 8000H~0FFFFH). When SYS_EX is used for A16, the usage of A16 is similar with A15. In addition, the content of PCI expansion ROM is copied to the ROM memory by BIOS, so setting the expansion ROM base address to remap U3 to memory space. A15 pin of CH365 is not only used as the address bus but also can be freely controlled. For example, when it is necessary that CH365 is connected with SRAM and ROM at the same time, the chip selection of the double is switched through A15.

The expansion ROM in the PC is equal to an electron disk. If the boot program and application program are written in it, those programs can control the computer to realize some functions even if the computer has not

hard disk and OS. For example, PC without hard disk is used for industrial control to control the peripheral device and the task process.

9.4. Connect to 2-wire Interface EEPROM (left figure below)

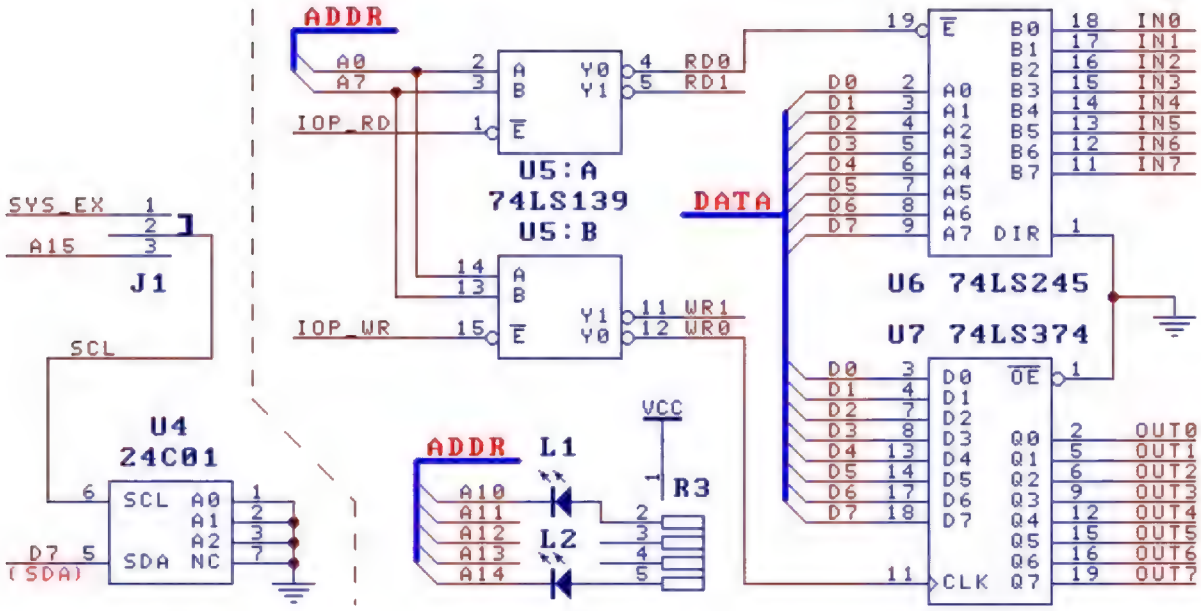
The signal line SCL of CH365 can select SYS_EX or A15.The default is A15 after system reset. SYS_EX is chosen while A15 is address bus, otherwise, choose A15. SDA signal line also is data bus D7.In order to prevent unnecessary mis-operation due to SDA changes during SCL at high. Usually, SCL (SYS_EX or A15) keeps low after system reset by the operation mode setting.

The 2-wire serial interface of Ch365 uses 7-bit device address which can be connected with multi-devices. The bit7 ~1 of the device address and command register are the 7-bit device address bits, which are used for selecting the slave device, and the bit 0 is the command bit, it is write operation when bit 0 is 0 and read operation when it is 1. For example, CH365 is connected with two 24C02 chips. The operation steps that reading from one chip and writing to another chip are the following:

Operation	Read from 12H	Write to 34H	Instruction
Examine A2-A1-A0 of 24C02	A2-A1-A0=000	A2-A1-A0=010	Actual device address
Device address and command register	Input 10100001	Input 10100100	Device address and command
Address set register	Input 12H	Input 34H	Set operation address
Data access register	Null	Input 56H	Input 56H
Control and status register	Set bit 0=0,others keep unchanged		Start interface operation
Control and status register	Wait for bit 0=0 or 10ms		Wait for operation completed
Data access register	Output 78H	Null	Output 78H

The C-language WDM and DLL program based on CH365 as follow:

```
CHAR    mByte;    //data unit, used to save data read from 24C02 or to be written data
CH365ReadI2C (0x50, 0x12, &mByte);    // read a byte from 24C02 address 12H of A2-0=000
CH365WriteI2C (0x52, 0x34, 0x56);    // write 56H to 24C02 address 34H of A2-0=010
```



9.5. I/O Application (right figure above)

Read strobe/enable IOP_RD and write strobe/enable IOP_WR control the decode enabling of 74LS139. 74LS139 makes that the address decoding outputs 2 paths of reading control and 2 paths of writing control. Through the 74LS245 inputs buffer and 74LS374 latch output to get 2 groups with 8-bit buffer input in each

group and 2 groups with 8-bit latch output in each group. For example, if the I/O base address of CH365 is set to 5A00H, then reading the 5A00H port means reading the first group of buffer input, and writing 5A01H port means writing the second group of latched output. If CH365 isn't connected with expansion ROM or memory, the idle address bus A14~ A10 and A15 can be used directly as the output control bus. After system reset, A14 ~A10 are low. A15 is low during system reset, after system reset, A15 is set by the operation mode and the default is high. Unless D0 is pulled down when system reset, A15 is set to low.

The input and output signals of CH365 are compatible with TTL and CMOS level which can connect with ADC/DAC/MCU chip and so on. The drive current of output pins is more than 5mA, and it can drive the LED after connecting with a current-limiting in series. CH365 provides the address bus A7-A0 for I/O address decoding. The effective offset address range is 0EFH~00H and the length is no more than 240 bytes. Generally, external circuit does not require chip selection or directly forced.

The C-language WDM and DLL program based on CH365 for above operations as follow:

```
UCHAR    mByte;
//data unit, used to save data read from I/O port or to be written to I/O.
mPCH365_IO_REG     mIoBase;
// I/O port base address; the actual data unit address is equal to the base address plus the offset address.
CH365GetIoBaseAddr( &mIoBase );
// get the base address of I/O port, which is optional operation and not need to be executed.
// if not getting the I/O base address, only appointing the offset address in the I/O operation, which means that the I/O base address is 0.
// after calling the DLL of CH365, the DLL will automatically add the offset address to the base address and then perform I/O operation.
// memory is similar to this, if only appointing the offset address in the memory operation, the DLL will automatically add the memory base address
CH365ReadIoByte( & mIoBase -> mCh365 IoPort [0x00], &mByte );
// in the above operation, read one-byte data from the 00H offset address of the I/O port, namely, read the buffer input of first group.
CH365WriteIoByte( & mIoBase -> mCh365 IoPort [0x01], 0x47 );
// In the above operation, write the data 47H into the 01H offset address of the I/O port, namely, as the latch output of second group.
CH365SetA15_A8( 0x24 ); // Set A13 at high, A10 at high and others at low.
```

In DOS or PC without OS, the assembly program for the above operation is:

```
MOV   AX, 0B109H                          ; read the PCI configuration space in the unit of character
MOV   BX, CH365_PCI_BUS_DEV_ADDR          ;PCI address of CH365 board card, namely, bus/device/function number
MOV   DI, 0010H                           ; the offset address PC_BASE_ADDR0 of I/O port base address register
INT   1AH                                 ; read the base address of I/O port, automatically set it when the computer is initialized
AND   CX, 0FFFEH                          ; get the base address of I/O port; the lowest bit is indication bit, shielded
MOV   BX, CX                              ; this value is the base address of I/O port
LEA   DX, [BX].CH365_ IO_PORT [0]         ; I/O port address of the buffer inputs of first group, I/O base address plus 0
IN    AL, DX                              ; read the buffer input data of 74LS245
LEA   DX, [BX].CH365_ IO_PORT [1]         ; I/O port address of latch output of the second group, I/O base address plus 1
MOV   AL, 47H
```

```
OUT     DX, AL                          ; write the data 47H into the 74LS374 latch output
                                          register
LEA     DX, [BX]. CH365_MEM_ADDR_H      ; A15-A8 address set register of I/O address, inside
                                          CH365
IN      AL, DX                          ; for maintaining the status of the other pins, firstly
                                          read the original A15-A8 address
OR      AL, 20H                         ; only set A13 at high, the other pins keep unchanged
AND     AL, 0F7H                        ; only set A11 at low, the other pins keep unchanged
OUT     DX, AL                          ; write new A15-A8 into address set register
```

9.6. Connect To MCU, etc.

PC makes bi-directional data transmission through CH365 and MCU or DSP. There are four ways: Firstly, using the dual-port SRAM to enable CH365 and MCU to write and read the same memory, and the bi-directional data exchange is made in the unit of big data blocks; secondly, using the bi-directional buffer interface chip CH421 to provide a 64-byte buffer for CH365 and MCU to write to each other respectively, and the bi-directional data exchange is made in the unit of 64-byte data blocks; thirdly, using the 8255 to provide asynchronous data exchange, and the bi-directional data exchange is made in the unit of one-byte data. Fourthly, the low speed transmission in the unit of half-byte or one-byte without additional hardware costs. For example, using the 4-bits data exchange interface with cooperated software, the 2-wire interface provided by CH365 itself and the SPI interface of software simulation.

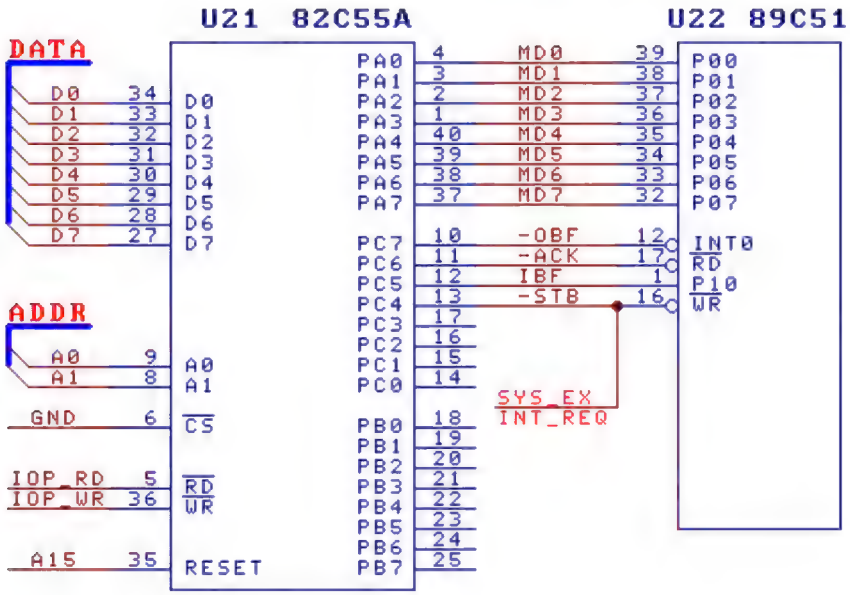
9.6.1. Asynchronous Data Exchange With Handshake Signal (below figure)

If A port works at the second mode, the I/O expansion chip 8255 provides the standard asynchronous data exchange with the handshake signal in the unit of byte. If the port A of 8255 is connected with CH365, MCU will control the 8225 and if it is connected with MCU, PC will control the 8255 through CH365. When the port A is working at the second mode, 8255 provides input buffer area full or empty signal, output buffer area full or empty handshake signal and interrupt signal for CH365 and MCU.

In the figure the port A of U21 (82C55A) is connected with the MCU U22 (89C51). The address bus A15 is used for resetting U21. The PC program sets the port A of U21 at the second mode. Low level triggers the MCU interrupt.

After PC writes data to MCU, the operation makes the –IBF to low and U22 enters the interrupt program. After MCU reads the data from the interrupt program, the read operation –ACK makes –OBF to high. The PC program has queried –0BF status, re-write data to MCU.

After MCU writes data to PC, the write operation –STB makes IBF to high. If enabling the interrupt function of CH365, low-level pulse is outputted from -STB to SYS_EX which makes PC enters the interrupt program. If the interrupt function is not enabled, the PC checks the IBF status by query method. The PC program has queried IBF status and reads the MCU data, the read operation –ACK makes IBF to low. MCU has queried the IBF status and re-write data to PC.



9.6.2. Use Bi-directional Buffer Interface Chip CH421

CH421 can provide the fast connection between CH365 and MCU. The detail content refer to the CH421 Datasheet.

9.7. I/O Expansion (refer to the above figure, remove U22)

CH365 can directly connect with most common I/O expansion circuit. In the figure U21 is used for I/O expansion. The read and write speed of previous 8255 is above 300ns, while the latter 82C55A can reach under 200ns, so U21 should choose the 82C55A. Address of 8255 port is located in offset address 00H~0EFH. In actual, only 03H~00H are used, and others are aliases. 8255 expands the 8-bit data bus to three 8-bit data ports. For example, writing the control character 10010000B to the offset address 03H, the PA port of U21 is set as 8-bit input, then the PB and PC are set as 8-bit output.

The C-language WDM and DLL program based on CH365 for above operations as follow:

```
UCHAR     mByte;
// data unit，used to save data read from the I/O or to be written to I/O
mPCH365_IO_REG     mIoBase = NULL;
// I/O base address, the DLL will automatically plus the base address when it is set to 0
CH365ReadIoByte( & mIoBase -> mCh365IoCtrl, &mByte );
// firstly read chip control register
CH365WriteIoByte( & mIoBase -> mCh365IoCtrl, mByte | mBitAddr15Out );
// A15=1
CH365WriteIoByte( & mIoBase -> mCh365IoCtrl, mByte & ~ mBitAddr15Out);
// A15=0
// the above three operations control A15 outputs the high pulse, thereby reset U21(82C55A).
CH365WriteIoByte( & mIoBase -> mCh365IoPort[0x03], 0x90 );
  // 10010000B
// the above operations set control character of 8225, PA input, PB/PC output.
CH365ReadIoByte( & mIoBase -> mCh365IoPort[0x00], &mByte );
// the above operations input data from PA of 8225
CH365WriteIoByte( & mIoBase -> mCh365IoPort[0x01], 0x8E );
// the above operations output 8EH to PB of 8225
```

9.8. Other Auxiliary Functions

9.8.1. Hardware Cycle Count Register

CH365 provides a hardware timing unit. The timing is inputted based on the 128 frequency division of PCI bus basic frequency. For the standard 33.3MHz basic frequency of PCI bus, one count is increased for the hardware cycle count register every 3.84uS. The time from 00H to FFH then from FFH to 00H is totally 983.04us. By comparing the difference between two counts, the actual delay can be calculated to replace the computer software command cycle with large error.

9.8.2. 8-bit Bus Input Port

CH365 provides 8-bit bus status input port in the PCI device configuration space. When reading the port, the read and write strobe control line of I/O and memory don't output signal and can read the static value of the local data bus D7~D0. If the data bus D6 is connected with the pull-down resistor, then bit 6 is 0, otherwise, bit 6 is 1. Generally, the function is used to identify the board card by application program or read the external configuration information of this board card by Expansion ROM boot program. For example, D6 and D7 are pulled-down which represents one configuration mode and only D6 is pulled-down which represents another mode. The application program or boot program can distinguish it according the pull-down.

9.9. Hardware Interrupt

CH365 supports interrupt request with active low. If data line D3 connects pull-down register, the operation setting is enable interrupt function. SYS_EX is multiplexed as the local interrupt request input pin INT_REQ.

After the INT_REQ pin has detected a low level, the interrupt active status bit of CH365 (bit 2 of the chip status register) is automatically set to 1. CH365 applies for interrupt to PCI bus by PCI_INTA. At that time, even if the INT_REQ resumes to high, CH365 keeps the interrupt active status until CH365 interrupt service program of PC clears the interrupt active state bit to 0. But CH365 enters the interrupt active state once again if INT_REQ still has detected the low level after clearing the interrupt active state bit.

If the PC program sets the interrupt activation status bit of CH365 to 1 in the software mode, it can also enable that CH365 enters the interrupt active status, and apply for interrupt to PCI bus. Such software interrupt has exactly the same characteristics as the hardware interrupt caused by the low level of the external INT_REQ, and it can be used to test the interrupt function of CH365.

For details, please refer to CH365EVT<CH365 Interrupt Function Instruction>. The standard interrupt process is as follows:

① The peripheral circuit outputs interrupt request signal with active low to the INT_REQ, which causes the interrupt active state bit changed to 1. If the peripheral circuit provides the low pulse, the pulse width must be longer than TINT.
② CH365 applies for interrupt to PC through the INTA signal of PCI bus.
③ PC accesses to the interrupt service of CH365.
④ If the interrupt service provided by peripheral circuit is not the pulse signal, the interrupt service should notify the peripheral circuit to cancel the interrupt inquest, that is INT_REQ resumes to high.
⑤ Interrupt service clears the interrupt active state bit of the chip control register of CH365 to 0. PC_INTA of CH365 resumes high-impedance status and cancels the interrupt inquest to PC.
⑥ Interrupt service does the necessary process. This step can be executed firstly after entering the interrupt.
⑦ After the interrupt process is completed, the PC exits the interrupt service of CH365.

9.10. Local Hardware Fixed Address and Update ISA (suit to update I/O card of ISA)

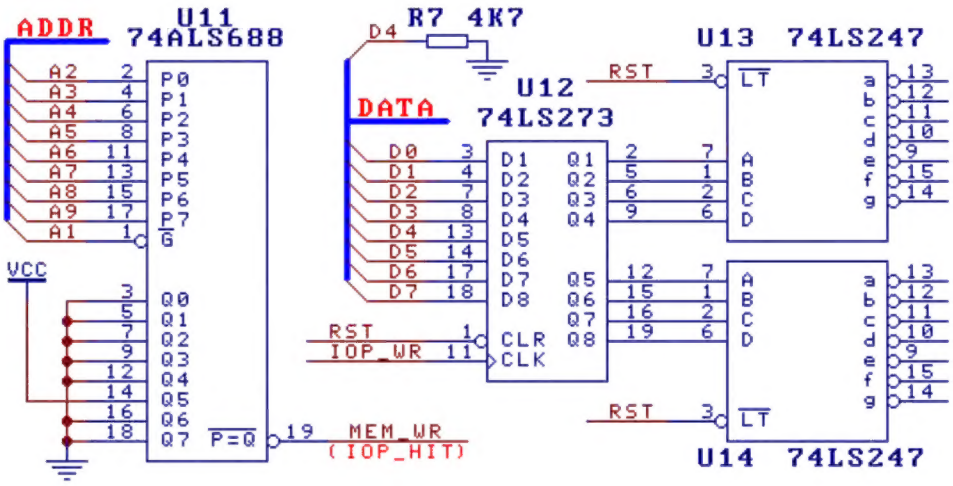
Generally, the I/O base address of PCI device is automatically attributed when PC is initialized. It ensures

that some special I/O addresses will not be occupied at will and I/O address of many devices will not conflict. This is different from ISA I/O address which can be used flexibly. PCI bus specification hasn't provided a method that defining the I/O address of PCI board by the product manufacture. This will be little inconvenient in the special application: Firstly, on different types of PC, the automatically attributed I/O address of the same PCI board may be different. Secondly, the card cannot work before the PC is initialized and the I/O address has not been attributed. Thirdly, the automatically attributed I/O address is above 1000H and cannot be located in the address range of 3FFH~ 000H and special I/O addresses. Finally, when ISA board is directly updated to PCI bus, need to modify application to first get the base address of I/O port before I/O operation.

CH365 provides a method that the I/O address of PCI board is selected by the product manufacture, which is local hardware fixed address. The theory is that some I/O address decoding of PCI device are realized by the 2-level peripheral circuit that is simple and similar with ISA I/O address decoding. CH365 synchronously provides the address of PC I/O operation to peripheral circuit. After the peripheral circuit matches the address decoding, CH365 requests the local hardware fixed address, then CH365 requests PCI to perform I/O operation at this address.

Taking the debug card (post card) based on PCI bus as an example. In the process of PC self-inspection, PC continuously outputs the post code to the I/O port of 0080H address. Since PC only attributes the I/O address for PCI after completing most of the self- inspection, the common PC card can't get the previous post code, even after attributing the I/O address, the common PCI board can't locate in the special I/O address 0080H.

The below figure is a part circuit of CH365 debug card. Resistor R7 is used to the operation mode setting. The data line D4 is pull-down and the value is 11101111. CH365 sets the multiplexed pin MEM_WR as local hardware fixed address request signal line IOP_HIT. The comparator U11 (use 74F138 to achieve) is used to match and compare the address A9~A1. When the address provided byCH365 is same with the preset address 0080H~ 0081H, U11 outputs low. The valid IO_RD read data or the valid OP_WR write data is outputted after CH365 has received local hardware fixed address request. For debug card, only the I/O data need to be received, data trigger U12 latches the post code, which is decoded by the character decoding circuit U13 and U14 and displays the data on the digital tube.



CH365 has limited the local hardware fixed address range 03FFH~0000H, which is correspond to the I/O address range of ISA. When PC is performing I/O operation, A9~A0 of CH365 synchronously outputs the operation address, the peripheral 2-level decoding circuit only need to match and compare with A9~A0. CH365 demands the delay time of 2-level decoding is no more than 20nS, in another word, the time that the peripheral circuit generates the local hardware fixed address request after decoding and comparing the address is no more than 20nS. The above figure shows the function by equivalent comparator 74AL688.The actual peripheral decoding circuit usually chooses the compiler 16V8or decoder 74F138.It is recommended

that 16V8 is 15nS.

The left below figure shows that the 2-level decoding circuit is achieved by TTL logic circuit. Because of the low speed of TTL logic circuit, the decoding time may exceed 20nS, the figure is only used for function and logic description. In actual, the circuit is recommended to use the right circuit which uses the simple compiler for performing 2-level decoding.

U18 decodes the A9~A4 of CH365 and generates eight 16-byte I/O address areas. If the original ISA card uses the address 270H~27FH, then Y7 pin of U18 directly connects the IOP_HIT pin of CH36. When PC performs the I/O operation in the range of 270H~27FH, IOP_HIT is set to low by U18.CH365 outputs the read and write strobe signal by IOP_RD and IOP_WR. The external circuit exchanges data with PC according to the read and write signal. Of course, the external circuit can further decodes A3-A0.

The actual circuit is recommended to use compiler U20 (GAL16V8) for the 2-level decoding, the circuit is simple and easy-to-use. In general, 16V8 can use the remaining resource (1pin output 1, 11 pin input, 15-19 pin output) for other function after completing the 2-level decoding. For example, for further address decoding, combining the low-bit address with IOP_RD and IOP_WR to perform the I/O selection to generate chip selection signals CS1~ CS5 and so on.

In addition, after using the local hardware fixed address function, CH365 not only responds the I/O operation in the hardware fixed address range, but also responds I/O operation in the I/O address range attributed automatically by PC. For example, the hardware fixed address range is 240H~247H and the I/O address range attributed automatically by PC is C000H-C0FFH, then operating address 240H is equal to the C000H, and reading or writing address 247H is equal to the C007.

Based on the local hardware fixed address function, the produce manufacture can directly update the original ISA card to PCI bus, and also can use the original I/O address without modify the application.

9.11. Set PCI ID

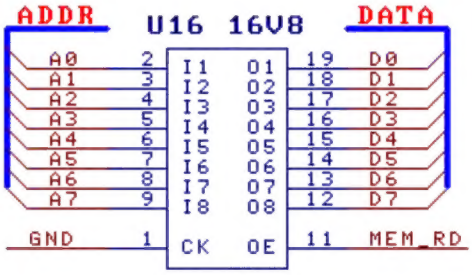
Generally, PCI card directly uses the default ID of CH365. If the manufacture needs to set their own device ID or special device ID, it can pull-down the data bus D1 after the system reset, and set the operation mode of CH365 to work in the external ID mode and provide the new device ID by peripheral circuit.

After the operation mode is set and CH365 uses the external ID. All units whose register attribute is S in the CH365 configuration space will be directly mapped to the local memory with 0040H as the initial address. In the other word, the configuration space address 3FH~00H is correspond to the local memory address 7FH~40H. For example, reading the configuration space address 00H is equal to the local memory address 0040H. Similarly, the 2CH is 006CH. If the data of local memory address 0043H~0040H is set to 12345678H, then the Vendor ID of PCI board is 5678H and the device ID is 1234H. Similarly, it can be set the subsystem ID of PCI board in the memory address 006FH~006CH.

The local memory that provides the new device ID includes he expansion ROM and other read-only memory. For the PCI board that supports the expansion ROM, it can configure PCI device ID in the address 007FH-0040H of the ROM chip without adding any peripheral components. If PCI board itself doesn't need

the Expansion ROM, but it need to set PCI ID, there are two methods: One is to add an extra ROM chip, which is only used to provide a new PCI ID, not as expansion ROM. The other one is to refer to the below circuit, which uses a simple compiler (such as 16V8) to simulate ROM chip and provides new PCI ID to CH365.

In the below figure, the tri-status output of U16 is disenabled when the MEM_RD is high. U16 outputs related data of Address A7-A0 when the MEM_RD is active low. For example, when the MEM-RD is active low if A7-A0 are 48H, then output PCI Revision ID, and A7-A0 are 49H, output PCI Class Code.



9.12. I/O Read And Write Memory

Generally, the local memory of CH365 is mapped to the PC memory space. Reading and writing of the local memory is processed in the memory space. For supporting the expansion ROM with larger capacity or memory and addressing operation for ease under DOS, CH365 also provides a method that memory space switches to I/O space, which is suitable for expansion ROM with larger capacity and memory with orderly data access. This method directly supports 64KB memory or expansion ROM. The step of reading and writing memory space through I/O are shown in the following table.

Register	Register Operation	Read Data Program(ASM)	Read Data Program(C/C++)

A15~A0 address set register ADRSR	Write into initial address	mov dx, PORT_ADDR
		mov ax, START_ADDR
		out dx, ax
			outport (PORT_ADDR, START_ADDR);

Memory data access register MEMDR	Read and write data sequentially, address increased automatically	mov dx, PORT_DATA
		mov di, BUFFER_ADDR
		mov cx, LENGTH
		rep insb
			int i;
			char buf [LENGTH];
			for (i=0; i<LENGTH; ++i)
			buf [i]=inportb (PORT_DATA);

Constants and variables used in the program:

PORT_ADDR represents the port address of the A15-A0 address set register (which is I/O base address + 0F0H);

PORT_DATA represents the port address of the memory data access register (which is I/O base address + 0F3H);

START_ ADDR represents the initial address of access data in the expansion ROM or memory;

LENGTH represents the length of access data which is number of bytes;

BUFFER_ADDR represents the initial address of buffer for storing read data; buf is data buffer.